# 8.AngularJS 服务(Service)

AngularJS 中你可以创建自己的服务，或使用内建服务。

## 什么是服务？

在 AngularJS 中，服务是一个函数或对象，可在你的 AngularJS 应用中使用。

AngularJS 内建了30 多个服务。

有个 **$location** 服务，它可以返回当前页面的 URL 地址。

[8-1]

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <script src="js/angular.min.js"></script>  
</head>  
<body>  
  
<div ng-app="myApp" ng-controller="myCtrl">  
 <p> 当前页面的url:</p>  
 <h3>{{myUrl}}</h3>  
</div>  
  
<p>该实例使用了内建的 $location 服务获取当前页面的 URL。</p>  
  
<script>  
 var ***app*** = angular.module('myApp', []);  
 ***app***.controller('myCtrl', function($scope, $location) {  
 $scope.myUrl = $location.absUrl();  
 });  
</script>  
  
</body>  
</html>

注意 **$location** 服务是作为一个参数传递到 controller 中。如果要使用它，需要在 controller 中定义。

## 为什么使用服务?

在很多服务中，比如 $location 服务，它可以使用 DOM 中存在的对象，类似 window.location 对象，但 window.location 对象在 AngularJS 应用中有一定的局限性。

AngularJS 会一直监控应用，处理事件变化， AngularJS 使用 **$location** 服务比使用 **window.location** 对象更好。

### **$location vs window.location**

|  |  |  |
| --- | --- | --- |
|  | **window.location** | **$location.service** |
| 目的 | 允许对当前浏览器位置进行读写操作 | 允许对当前浏览器位置进行读写操作 |
| API | 暴露一个能被读写的对象 | 暴露jquery风格的读写器 |
| 是否在AngularJS应用生命周期中和应用整合 | 否 | 可获取到应用生命周期内的每一个阶段，并且和$watch整合 |
| 是否和HTML5 API的无缝整合 | 否 | 是（对低级浏览器优雅降级） |
| 和应用的上下文是否相关 | 否，window.location.path返回"/docroot/actual/path" | 是，$location.path()返回"/actual/path" |

## $http 服务

**$http** 是 AngularJS 应用中最常用的服务。 服务向服务器发送请求，应用响应服务器传送过来的数据。

使用 $http 服务向服务器请求数据:

[8-2]

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <script src="js/angular.min.js"></script>  
</head>  
<body>  
  
<div ng-app="myApp" ng-controller="myCtrl">  
  
 <p>欢迎信息:</p>  
  
 <h1>{{myWelcome}}</h1>  
  
</div>  
  
<p> $http 服务向服务器请求信息，返回的值放入变量 "myWelcome" 中。</p>  
  
<script>  
 var ***app*** = angular.module('myApp', []);  
 ***app***.controller('myCtrl', function($scope, $http) {  
 $http.get("welcome.htm").then(function (response) {  
 $scope.myWelcome = response.data;  
 });  
 });  
</script>  
  
</body>  
</html>

以上是一个非常简单的 **$http** 服务实例，更多 **$http** 服务应用请查看 [AngularJS Http 教程](https://www.runoob.com/angularjs/angularjs-http.html)。

## $timeout 服务

AngularJS **$timeout** 服务对应了 JS **window.setTimeout** 函数。

两秒后显示信息:

[8-3]

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <script src="js/angular.min.js"></script>  
</head>  
<body>  
  
<div ng-app="myApp" ng-controller="myCtrl">  
  
 <p>两秒后显示信息:</p>  
  
 <h1>{{myHeader}}</h1>  
  
</div>  
  
<p>$timeout 访问在规定的毫秒数后执行指定函数。</p>  
  
<script>  
 var ***app*** = angular.module('myApp', []);  
 ***app***.controller('myCtrl', function($scope, $timeout) {  
 $scope.myHeader = "Hello World!";  
 *//$timeout 服务* $timeout(function () {  
 $scope.myHeader = "How are you today?";  
 }, 2000);  
 });  
</script>  
  
</body>  
</html>

## $interval 服务

AngularJS **$interval** 服务对应了 JS **window.setInterval** 函数。

每一秒显示信息:

[8-4]

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <script src="js/angular.min.js"></script>  
</head>  
<body>  
  
<div ng-app="myApp" ng-controller="myCtrl">  
  
 <p>现在时间是:</p>  
  
 <h1>{{theTime}}</h1>  
  
</div>  
  
<p>$interval 访问在指定的周期(以毫秒计)来调用函数或计算表达式。</p>  
  
<script>  
 var ***app*** = angular.module('myApp', []);  
 ***app***.controller('myCtrl', function($scope, $interval) {  
 $scope.theTime = new ***Date***().toLocaleTimeString();  
 *//$interval 服务:间隔指定毫秒数执行一次* $interval(function () {  
 $scope.theTime = new ***Date***().toLocaleTimeString();  
 }, 1000);  
 });  
</script>  
  
</body>  
</html>

## 创建自定义服务

你可以创建自定义服务，链接到你的模块中：

创建名为hexafy 的服务:

app.service('hexafy', function() {  
 this.myFunc = function (x) {  
 return x.toString(16);  
 }  
 });

要使用自定义服务，需要在定义控制器的时候独立添加，设置依赖关系:

使用自定义的的服务 hexafy 将一个数字转换为16进制数:

[8-5]

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <script src="js/angular.min.js"></script>  
</head>  
<body>  
<div ng-app="myApp" ng-controller="myCtrl">  
  
 <p>255 的16进制是:</p>  
  
 <h1>{{hex}}</h1>  
  
</div>  
  
<p>自定义服务，用于转换16进制数：</p>  
  
<script>  
 var ***app*** = angular.module('myApp', []);  
  
 *//service:自定义服务* ***app***.service('hexafy', function() {  
 this.myFunc = function (x) {  
 return x.toString(16);  
 }  
 });  
 ***app***.controller('myCtrl', function($scope, hexafy) {  
 $scope.hex = hexafy.myFunc(255);  
 });  
</script>  
  
</body>  
</html>

## 过滤器中，使用自定义服务

当你创建了自定义服务，并连接到你的应用上后，你可以在控制器，指令，过滤器或其他服务中使用它。

在过滤器 myFormat 中使用服务 hexafy:

[8-6]

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <script src="js/angular.min.js"></script>  
</head>  
<body>  
  
<div ng-app="myApp">  
 在过滤器中使用服务:  
  
 <h1>{{255 | myFormat}}</h1>  
  
</div>  
  
<script>  
 var ***app*** = angular.module('myApp', []);  
 ***app***.service('hexafy', function() {  
 this.myFunc = function (x) {  
 return x.toString(16);  
 }  
 });  
 *//过滤器中使用自定义服务* ***app***.filter('myFormat',['hexafy', function(hexafy) {  
 return function(x) {  
 return hexafy.myFunc(x);  
 };  
 }]);  
  
</script>  
  
</body>  
</html>

在对象数组中获取值时你可以使用过滤器：

创建服务 hexafy:

[8-7]

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <script src="js/angular.min.js"></script>  
</head>  
<body>  
  
<div ng-app="myApp" ng-controller="myCtrl">  
 <p>在获取数组 [255, 251, 200] 值时使用过滤器:</p>  
  
 <ul>  
 <li ng-repeat="x in counts">{{***x*** | myFormat}}</li>  
 </ul>  
  
 <p>过滤器使用服务将10进制转换为16进制。</p>  
</div>  
  
<script>  
 var ***app*** = angular.module('myApp', []);  
 ***app***.service('hexafy', function() {  
 this.myFunc = function (x) {  
 return x.toString(16);  
 }  
 });  
 ***app***.filter('myFormat',['hexafy', function(hexafy) {  
 return function(x) {  
 return hexafy.myFunc(x);  
 };  
 }]);  
 ***app***.controller('myCtrl', function($scope) {  
 $scope.counts = [255, 251, 200];  
 });  
</script>  
  
</body>  
</html>